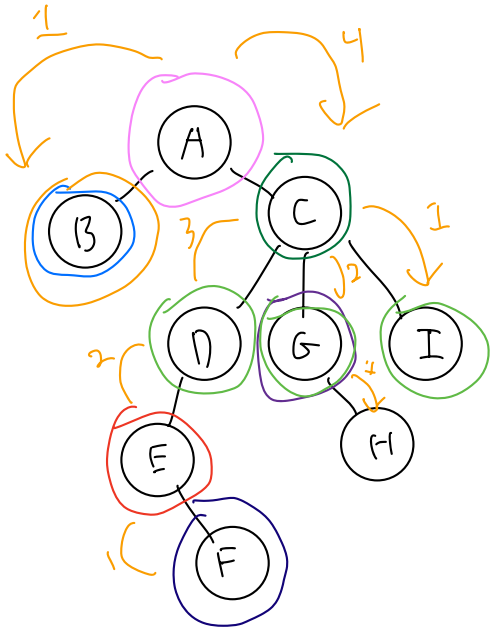
Trees!



root node : A

B : a child of A

E : a parent of F

D, G, I are siblings

F : a decendant of A, C, D, E

C : is an ancentor of F, H ... more

leaf / external nodes are : B, F, H, I

branch / internal nodes are : A, C, D, E, G

Depth of H : 3

Depth of F : 4

Height of tree : 4

tree Node

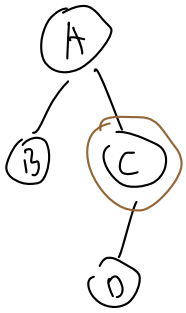
Parent

value

[list_head
  list_tail
  num children]

G
- C
- G
H
H
1

tree Node (binary)



Parent

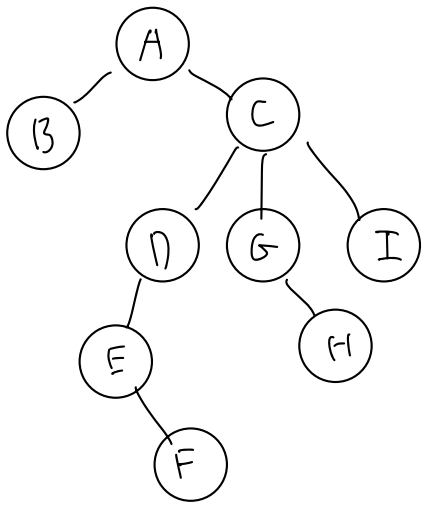
value

[left
  right]

C
- A
- C
[- D
  null]

Pre order

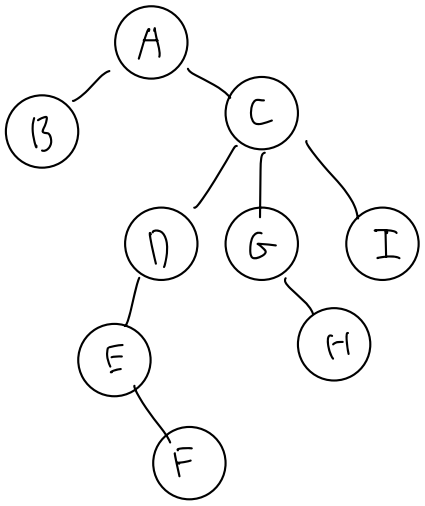
Print: A, B, C, D, E, F, G, H, I




```
void PreOrder(node) {
    if (node == nullptr){
        return;
    }
    Print (node->val)
    Preorder (node->child1)
    Pre order (node->child2)
    Pre ordre(node->child3)
        :
}
```

A B C D E F G H I

Post order



Print: B F E D H G I C A

```
void PostOrder(node) {
    if (node == nullptr){
        return;
    }
    Preorder (node->Child1)
    Pre order (node->Child2)
    Pre ordre(node->Child3)
    Print (node->val)
}
```